Подходы к оптимизации гиперпараметров в задаче кластеризации изображений в социальных медиа

И. В. Прусских

Санкт-Петербургский государственный университет
prusskih.ilay@gmail.com

В. Ф. Столярова, Ф. В. Бушмелев

Санкт-Петербургский Федеральный исследовательский центр Российской академии наук
[vfs,fvb]@dscs.pro

Аннотация. **Рост популярности социальных медиа, таких как ВКонтакте, влечет за собой увеличение объема графических цифровых следов, оставляемых пользователями, например, аватаров. Данная информация помогает исследователям использовать модели машинного обучения для разработки решений в различных областях: маркетинге и банковском скоринге, охране общественного здоровья и информационной безопасности. Современные подходы являются гибкими и универсальными инструментами, которые требуют адаптации к особенностям конкретных наборов данных. В данной работе рассматривается задача подбора гиперпараметров с целью повышения качества кластеризации аватаров пользователей ВКонтакте. Алгоритм, лежащий в основе данного исследования, является распространенным для анализа изображений и состоит из последовательного применения нейросетевой модели излечения признаков, шага снижения размерности, выделения выбросов и непосредственно шага кластеризации. В работе рассматривается последовательность методов CLIP–UMAP–LoF–HDBSCAN. В качестве оптимизаторов описанных методов рассматривались решётчатый обход (Grid Search), случайный поиск (Random Search), оптимизация на основе градиента (Gradient-based optimization), байесовская оптимизация (Bayesian Optimization) и эволюционный подход (Evolutionary optimization). При проведении эксперимента был собран набор данных, состоящих из более чем 50 тысяч аватаров пользователей ВКонтакте, в результате которого наиболее эффективным методом для кластеризации аватаров пользователей оказалась байесовская оптимизация, позволившая получить устойчивые кластеры со значением силуэтов более 0,56. Также были получены значения гиперпараметров, которые позволяют получить наилучшую кластеризацию сходных наборов данных.**

Ключевые слова: кластеризация; подбор гиперпараметров; CLIP; снижение размерности; локальный уровень выброса; байесовская оптимизация; социальные медиа

I. Введение

В последние десятилетия неотъемлемой частью современных технологий стали нейросетевые модели, активно применяемые в самых различных областях – от обработки изображений до генерации текстов [1, 2]. В различных задачах анализа данных нейросетевые подходы используются при получении эмбеддингов или численных представлений семантических характеристик различных объектов (изображений или текстов).

Эти подходы применяются и при анализе цифровых следов пользователей онлайн медиа [3, 4]. Одной из задач в этой сфере является выявление групп семантически схожих изображений [5], при этом для выделения их характеристик используются энкодеры, которые позволяют представить смысловую информацию в виде числового вектора высокой размерности. Для применения современных методов кластеризации [6] используются методы снижения размерности полученных эмбеддингов. При этом и методы снижения размерности, и методы кластеризации зависят от некоторого числа внутренних параметров (гиперпараметров), которые оказывают значительное влияние на качество получаемого разбиения данных.

Отметим также, что сама кластеризация зависит от типа кластеризуемых изображений, и для аватаров имеет свои особенности, как, например, наличие большого класса с изображением лиц [7]. Для определения аномалий и повышения качества кластеризации может использоваться дополнительный шаг – определение значения локального уровня выброса (LOF) [8]. Этот шаг позволяет снизить количество наблюдений и получить более четкие кластеры. Возникает актуальная задача определения сочетания гиперпараметров методов (настройки) методов снижения размерности и определения аномалий, [9], которое позволяет получить наилучшую кластеризацию и учитывает особенности датасета [10].

Данная задача чаще всего решается при помощи эвристик или же перебором наборов значений по заранее заданной сетке, однако эти методы являются трудозатратными. Целью исследования является выявление особенностей процесса определения гиперпараметров методов UMAP–LOF, позволяющих получить оптимальную кластеризацию изображений-аватаров пользователей онлайн социальной сети методом HDBSCAN, обработанных с помощью автоэнкодера CLIP. Практическая значимость исследования заключается в определении оптимальных комбинации параметров пайплайна UMAP–LOF для решения задачи выявления групп схожих по смыслу аватаров. Теоретическая значимость заключается в определении границ применимости методов

Работа выполнена при финансовой поддержке в рамках проекта по государственному заданию СПБ ФИЦ РАН № FFZF-2025-0006

оптимизации гиперпараметров для задачи кластеризации аватаров пользователей социальной сети с точки зрения качества получаемой кластеризации.

II. Материалы и методы

A. *Uniform Manifold Approximation and Projection (UMAP)*

UMAP – алгоритм нелинейного снижения размерности, часто используемый для визуализации сложных многомерных данных и анализа скрытых структур в них. Основное преимущество UMAP заключается в его способности сохранять локальную структуру данных в низкоразмерном пространстве [11].

Основные гиперпараметры UMAP, которые существенно влияют на результат кластеризации, включают: **n_neighbors**, определяющий число ближайших соседей для построения графа локальной плотности (чем выше значение, тем лучше выражена глобальная структура данных); **n_components**, задающий количество измерений после понижения размерности и влияющий на сохранение информации из исходных данных; **min_dist**, регулирующий минимальное расстояние между точками в низкоразмерном пространстве и тем самым влияющий на плотность кластеров; **metric**, определяющий метрику расстояния, выбор которой зависит от особенностей набора данных; и **spread**, контролирующий ширину кластеров в низкоразмерном пространстве, что позволяет управлять их распределением.

B. *Local outlier factor (LOF)*

LOF – метод определения выбросов данных, основанный на локальной плотности точек. Данный алгоритм сравнивает плотность точки с плотностью её соседей и на основе этого определяет коэффициент локального выброса [12].

Основные гиперпараметры LOF, влияющие на результат обнаружения выбросов, включают: **n_neighbors**, который определяет количество ближайших соседей для оценки локальной плотности точки (чем больше значение, тем менее чувствителен алгоритм к локальным выбросам); **metric**, задающий метрику расстояния, выбор которой зависит от особенностей набора данных; и **contamination**, указывающий долю выбросов в данных, если она известна, и влияющий на количество точек, помеченных как выбросы на основе их коэффициента локального выброса.

C. *Методы подбора гиперпараметров*

Далее будут приведены различные методы оптимизации гиперпараметров с выделением их плюсов и минусов [13, 14].

Решётчатый подход (Grid Search) является одним из наиболее распространенных и простых подходов решения проблемы настройки гиперпараметров. Данный метод предполагает задание наборов значений для каждого из гиперпараметров и последующий перебор этих значений по решётке для выявления наилучшей комбинации. Плюсом решётчатого подхода является, то, что он совершает полный перебор комбинаций гиперпараметров из данного набора, что гарантирует нахождение лучшей комбинации. Однако метод ресурсозатратен, что ограничивает его применимость.

Случайный поиск (Random Search) [15]. В отличие от решётчатого поиска, где перебираются все возможные комбинации, данный метод выбирает случайные наборы значений, что позволяет гораздо быстрее найти хороший набор гиперпараметров. Однако, так как данный метод перебирает лишь часть возможных комбинации, существует вероятность упустить лучший набор значений.

Байесовская оптимизаций (Bayesian Optimization) [16]. Данный метод основан на байесовском выводе для функции, оценивающей качество получаемой модели на основе используемых гиперпараметров. Байесовская оптимизация является наиболее популярным методом решения проблемы подбора гиперпараметров.

Эволюционный подход (Evolutionary optimization) [17]. Этот подход использует эволюционный алгоритм, вдохновленный биологической концепцией эволюции. Данный метод позволяют найти оптимальные значения гиперпараметров. Однако, он довольно сложно реализуем и ресурсозатратен из-за необходимости частой оценки эффективности и производительности модели.

D. *Кластеризация*

HDBSCAN [18] сочетает в себе методы плотностной кластеризации dbscan и иерархический подбор оптимальной кластеризации. В исследовании использовались значения параметра **min_cluster_size** равный 500 в силу объема имеющегося датасета.

Существует множество метрик кластеризации, которые отражают различные аспекты выявленных групп [19]. Одним из популярных решений является анализ ширины силуэтов, который отражает относительную схожесть объектов внутри отдельного кластера по сравнению с объектами других кластеров.

Значение силуэта для объекта i определяется по следующей формуле:

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

где a_i – среднее расстояние от объекта i до всех остальных объектов в кластере, b_i – среднее расстояние от объекта i до всех точек соседнего кластера. S_i принимает значения от -1 до 1.

Среднее значение силуэта по всем объектам набора данных называется коэффициентом силуэта и характеризует качество всей кластеризации. Чем ближе данное значение к 1, тем лучше разделены кластеры. При значениях коэффициента более 0.7 считают, что кластеризации «сильная», значениях более 0.5 – «приемлемая» и значениях более 0.25 – «слабая».

III. Вычислительный эксперимент

В данном разделе приведен эксперимент, использовавшийся для выявления наиболее

эффективного подхода к подбору гиперпараметров последовательности методов UMAP–LOF.

A. Реализация и используемые технологии

В качестве языка программирования был выбран Python. Одним из ключевых требований к процессу обработки данных, учитывая необходимость последующего подбора гиперпараметров, является высокая скорость его выполнения. Поэтому использованы GPU-реализации алгоритмов UMAP и HDBSCAN с помощью открытых библиотек RAPIDS[1], предоставляющий инструменты для выполнения всего цикла обработки данных и аналитики на GPU. Для реализации алгоритма LOF выбрана версия из sklearn[2], использующая CPU, в связи с отсутствием GPU версии во всех распространенных библиотеках. Для подобора гиперпараметров использовалась библиотека Optuna[3].

Все эксперименты проводились на персональном компьютере с процессор AMD Ryzen 5 5600X с 6 ядрами и видеокартой NVIDIA GeForce RTX 4070.

B. Набор данных

Данные для исследования были получены с помощью приложения, которое размещено на платформе VK MiniApps, предоставляющее возможности каждому пользователю онлайн социальной сети пройти ряд психологических тестов. При использовании приложения, пользователям предлагалось добровольное информированное согласие для использования данных их профилей в научных исследованиях. Для экспериментов использовался датасет, состоящий из более чем 50000 аватаров 7843 пользователей, представленных в виде эмбеддингов размерностью 767, получаемых после обработки аватаров с помощью CLIP (ViT-L-14).

C. Шаги вычислительного эксперимента[4]

1) Подбор гиперпараметров при помощи поиска по решётке: UMAP - n_components: [2, 5, 10, 15, 20, 35, 50], n_neighbors: [100, 250, 500, 1000, 1500, 2000], min_dist: [0.0, 0.025, 0.1, 0.25]; LOF - n_neighbors: [10, 100, 250, 500, 1000]. Вычисление ширины силуэтов. Выбор наилучшего значения.

2) Подбор гиперпараметров при помощи случайного поиска, байесовской оптимизации и эволюционного подхода с диапазонами значений параметров: UMAP - n_components от 2 до 50, n_neighbors от 100 до 2000, min_dist от 0.0 до 0.25 с шагом 0.01; LOF - n_neighbors от 10 до 1000. Каждому алгоритму дана 100 попыток подобрать значения. Вычисление ширины силуэтов и выбор наилучшего набора гиперпараметров на основе её значения.

3) Повторение шага 2 восемь раз для уменьшения влияния случайности на результаты эксперимента (в связи со стохастичностью алгоритмов).

4) Вычисление медианы и размаха наилучшего значения ширины силуэта для методов случайный поиск, байесовская оптимизация и эволюционный подход.

IV. Результаты

В табл. 1 представлены результаты вычислительного эксперимента. На основе полученных результатов можно сделать вывод, что все стратегии подбора гиперпараметров продемонстрировали способность находить в плане ширины силуэтов наборы гипермараметров. Наиболее эффективный набор параметров был найден методом grid search, однако из-за значительного увеличения времени выполнения этого метода по сравнению с другими стратегиями его нельзя считать оптимальным. Наилучшей стратегией в данном случае оказалась байесовская оптимизация, которая продемонстрировала стабильные результаты в поиске оптимальных наборов гиперпараметров. Случайный поиск также показал сопоставимые результаты и небольшой размах.

ТАБЛИЦА I. Медиана и размах коэффициентов силуэта при найденных алгоритмами гиперпараметрах

Метод подбора	**Медиана**	**Размах**	**Время поиска решения**
Grid search	0.575	—	~8 часов
Random search	0.538	0.051	~1.5 часа
Bayesian optimization	0.558	0.047	~1.5 часа
Evolutionary optimization	0.528	0.245	~1.5 часа

Наиболее качественный набор значений с коэффициент силуэтов равным 0.575 удалось найти методом grid search. Данное значение силуэтов означает, что удалось получить умеренно хорошую кластеризацию, это означает что кластеры различимы, но внутри кластеров могут быть области, где плотность объектов недостаточно высока. Пример такой кластеризации приведен на рис. 1. Стоит отметить, что на легенде графика рис. 1 выделены два кластера с «шумом», где «-1» обозначает шум, найденный на этапе кластеризации HDBSCAN, а «-2» шум от LOF.

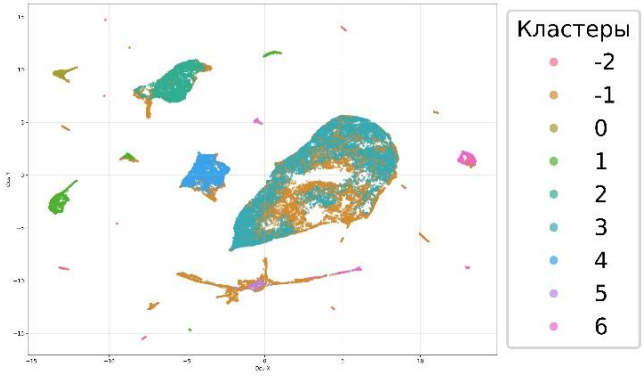

Рис. 1. Пример визуализации кластеризации аватаров пользователей (из 9 групп), которая получается для наилучшего набора значений гиперпараметров

[1] Team RAPIDS. RAPIDS: GPU Data Science and Machine Learning. 2024. — URL: https://docs.rapids.ai/ (дата обращения: 20.03.2025).

[2] Scikit-learn developers. Scikit-learn: Machine Learning in Python. 2024. — URL: https://scikit-learn.org/stable/api/sklearn.html (дата обращения: 20.03.2025).

[3] Optuna. Optuna: Hyperparameter Optimization Framework. 2024. — URL: https://optuna.readthedocs.io/en/stable/reference/index.html (дата обращения: 20.03.2025).

[4] С кодом можно ознакомиться по ссылке https://github.com/ilay631/hyperparameter_tuning.git

Значения гиперпараметров из найденного набора следующие:

- n_components (UMAP) = 20
- n_neighbors (UMAP) = 100
- min_dist (UMAP) = 0.0
- n_neighbors (LOF) = 1000

В табл. 2 представлены параметры 5 лучших кластеризаций в плане коэффициента ширины силуэта. Отметим, отметим, что для получения наилучшей кластеризации следует использовать снижение размерности до 9–20 компонент. Для датасетов, опирающихся на изображения-аватары, оптимальным является число соседей около 100–150 для UMAP и около 1000 для определения выбросов.

ТАБЛИЦА II. ПАРАМЕТРЫ ЛУЧШИХ КЛАСТЕРИЗАЦИЙ В ТЕРМИНАХ ШИРИНЫ СИЛУЭТОВ

#	UMAP			LOF	**Количество кластеров и размер шумового кластера (%)**	**Коэффициент силуэта**	**Метод подбора**
	n_components	n_neighbors	min_dist	n_neighbors			
1	20	100	0.0	1000	7 / 30%	0.575	Grid
2	15	100	0.0	1000	6 / 31%	0.573	Grid
3	15	154	0.0	872	7 / 35%	0.571	Bayes
4	9	154	0.0	916	7 / 37 %	0.568	Bayes
5	15	100	0.025	1000	6 / 34 %	0.567	Grid

Всего среди топ-50 по ширине силуэтов кластеризаций 16 были определены методом поиска по решетке и 33 – байесовским поиском.

Отметим, что для лучших вариантов кластеризации размер шумового кластера был значительным — порядка 30 %, что говорит о том, что около трети наблюдений не могут быть отнесены к выделенным кластерам. Это может быть связано с особенностями выявления смысловых сущностей при помощи нейросетевого подхода, необходимостью его донастройки.

V. ЗАКЛЮЧЕНИЕ

Исследование направлено на определение оптимального набора параметров классических методов обработки эмбеддингов UMAP и LOF, которые позволяют получить наилучшую кластеризацию аватаров в онлайн социальной сети. Такой набор данных обладает рядом особенностей: существует большой кластер с изображениями людей, и множество небольших кластеров, которые описывают иные сущности, которые обычно используют для аватаров. Было получено, что оптимальным является количество компонент UMAP большее 10, количество соседей от 100 до 154. Если говорить о количестве соседей при определении выбросов, то здесь оптимальным количеством соседей является число от 850 до 1000.

Кроме того, было получено, для решения задачи подбора оптимального сочетания параметров методы решетчатого поиска и байесовской оптимизации дают похожие по качеству кластеризации, однако время работы метода байесовской классификации меньше, что может быть важно для оптимизации ресурсов при работе с большими наборами данных.

Результаты исследования могут использоваться в различных задачах определения личностных особенностей (personality computing) при установлении взаимосвязей с графическими следами пользователей онлайн медиа.

СПИСОК ЛИТЕРАТУРЫ

[1] Archana R., Jeevaraj P. S. E. Deep learning models for digital image processing: a review //Artificial Intelligence Review. 2024. Т. 57. №. 1. С. 11.

[2] Minaee S. et al. Deep learning-based text classification: a comprehensive review //ACM computing surveys (CSUR). 2021. Т. 54. №. 3. С. 1-40.

[3] Gandhi A. et al. Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions //Information Fusion. 2023. Т. 91. С. 424-444.

[4] Oliseenko V. D. et al. Automating the temperament assessment of online social network users // Doklady Mathematics. Moscow: Pleiades Publishing, 2023. Т. 108. Suppl 2. С. S368-S373. DOI:10.1134/S1064562423701041

[5] Kim J., Kang Y. Automatic classification of photos by tourist attractions using deep learning model and image feature vector clustering //ISPRS International Journal of Geo-Information. 2022. Т. 11. №. 4. С. 245.

[6] Zhang H., Peng Y. Image clustering: An unsupervised approach to categorize visual data in social science research //Sociological Methods & Research. 2024. Т. 53. №. 3. С. 1534-1587.

[7] Bushmelev F., Stoliarova V., Tulupyeva T. Semantic Based Clusters of VK Users Avatars and Their Association with the Big Five Personality Profiles // International Conference on Intelligent Information Technologies for Industry. Cham: Springer Nature Switzerland, 2024. С. 183-192.

[8] Colomba L., Cagliero L., Garza P. Density-based clustering by means of bridge point identification //IEEE Transactions on Knowledge and Data Engineering. 2022. Т. 35. № 11. С. 11274-11287.

[9] Agrawal T. Hyperparameter optimization in machine learning: make your machine learning and deep learning models more efficient. New York, NY, USA: : Apress, 2021. 166 p.

[10] Chaudhry M. et al. A systematic literature review on identifying patterns using unsupervised clustering algorithms: A data mining perspective //Symmetry. 2023. Т. 15. №. 9. С. 1679.

[11] McInnes L., Healy J., Melville J. UMAP: Uniform manifold approximation and projection for dimension reduction // arXiv preprint arXiv:1802.03426. 2018.

[12] LOF: identifying density-based local outliers / M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander // Proceedings of the 2000 ACM SIGMOD international conference on Management of data. 2000. P. 93–104.

[13] Yang L., Shami A. On hyperparameter optimization of machine learning algorithms: Theory and practice // Neurocomputing. 2020. Vol. 415. P. 295–316.

[14] Feurer M., Hutter F. Hyperparameter optimization // Automated machine learning: Methods, systems, challenges. 2019. P. 3–33.

[15] Bergstra J., Bengio Y. Random search for hyper-parameter optimization // Journal of machine learning research. 2012. Vol. 13, № 2.

[16] Snoek J., Larochelle H., Adams R. P. Practical Bayesian Optimization of Machine Learning Algorithms // Advances in Neural Information Processing Systems / Ed. by F. Pereira, C. J. Burges, L. Bottou, K.Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012.

[17] Optimizing deep learning hyper-parameters through an evolutionary algorithm / S.R. Young, D.C. Rose, T.P. Karnowski et al. // Proceedings of the workshop on machine learning in high-performance computing environments. 2015. P. 1–5.

[18] McInnes L. et al. hdbscan: Hierarchical density based clustering //J. Open Source Softw. 2017. Т. 2. №. 11. С. 205.

[19] Everitt B., Landau S., Leese M., Stah D. Cluster analysis. 5-th edition. Wiley series in probability and statistics 848. 348 pp.